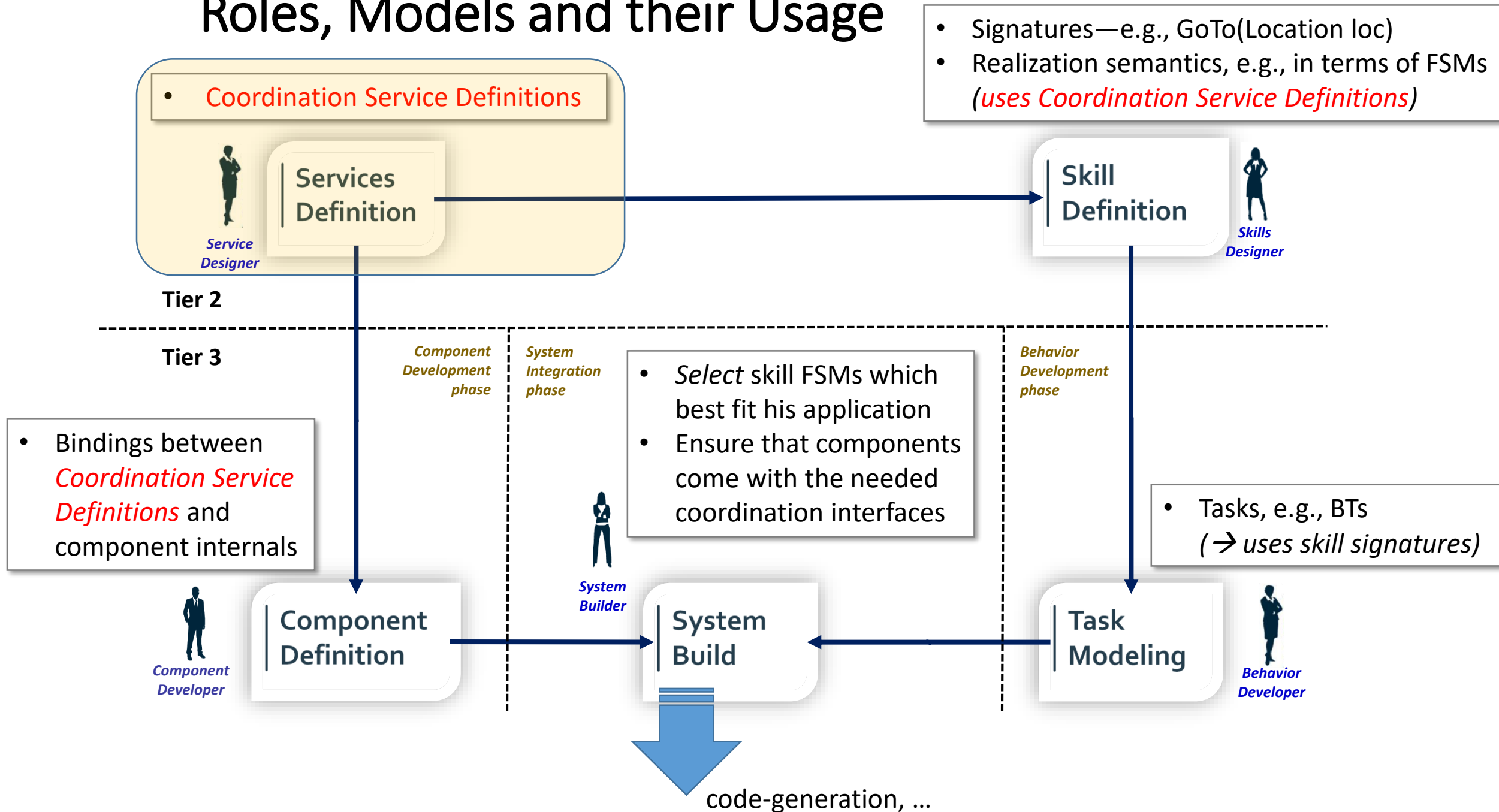


# *Skill Realizations in P4R*

*Modeling*

*P4R Team, 30/06/2020*

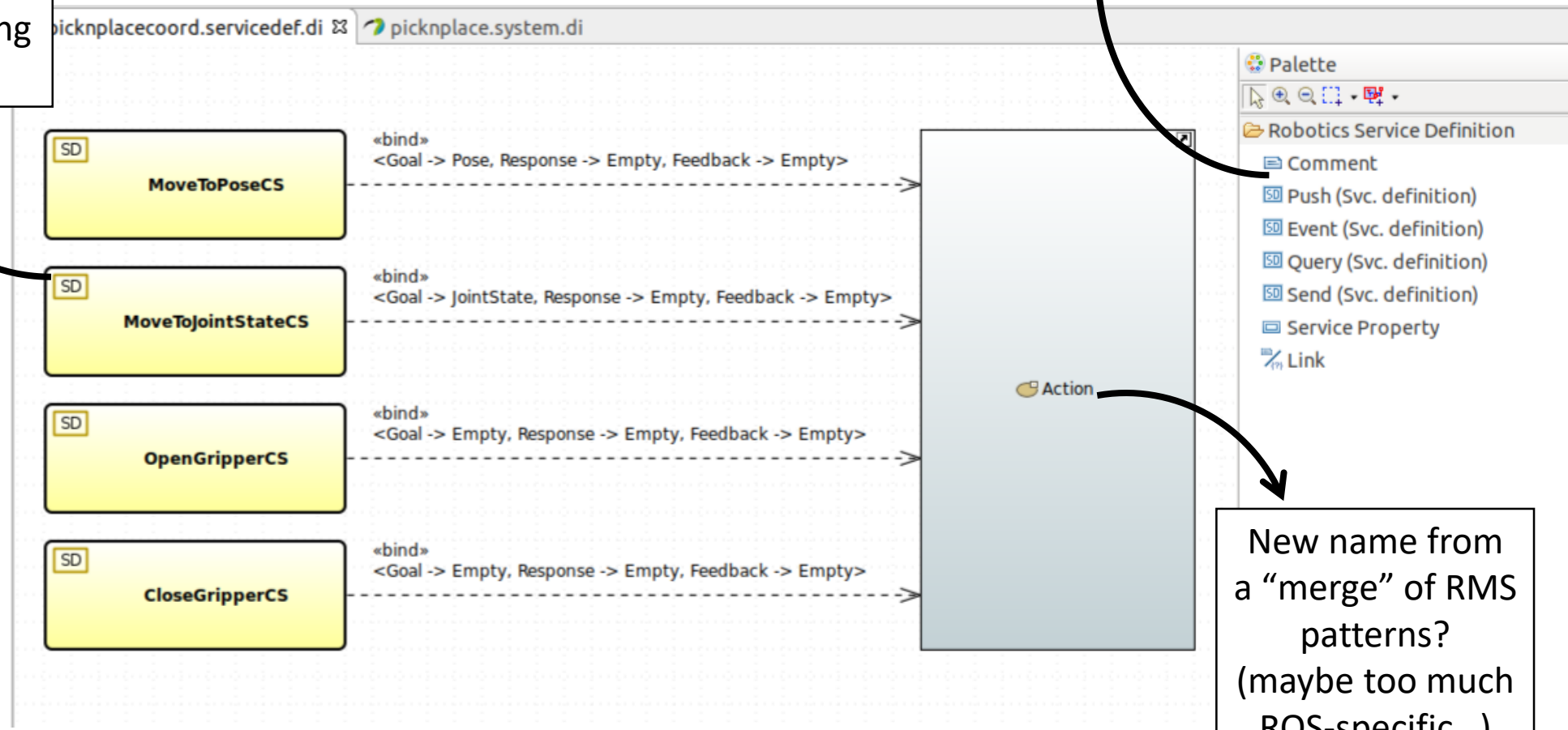
# Roles, Models and their Usage



# Coordination Service Definitions

CS label + new color  
(*new stereotype*  
→ custom modeling  
front-end)

New palette item "Coordination"



New name from  
a "merge" of RMS  
patterns?  
(maybe too much  
ROS-specific...)

# Roles, Models and their Usage

- **Coordination Service Definitions**



Services  
Definition

Tier 2

Tier 3

*Component  
Development  
phase*

*System  
Integration  
phase*

*Behavior  
Development  
phase*

- Bindings between **Coordination Service Definitions** and component internals



Component  
Definition



System  
Build

code-generation, ...

- Signatures—e.g., GoTo(Location loc)
- Realization semantics, e.g., in terms of FSMs (*uses Coordination Service Definitions*)

Skill  
Definition

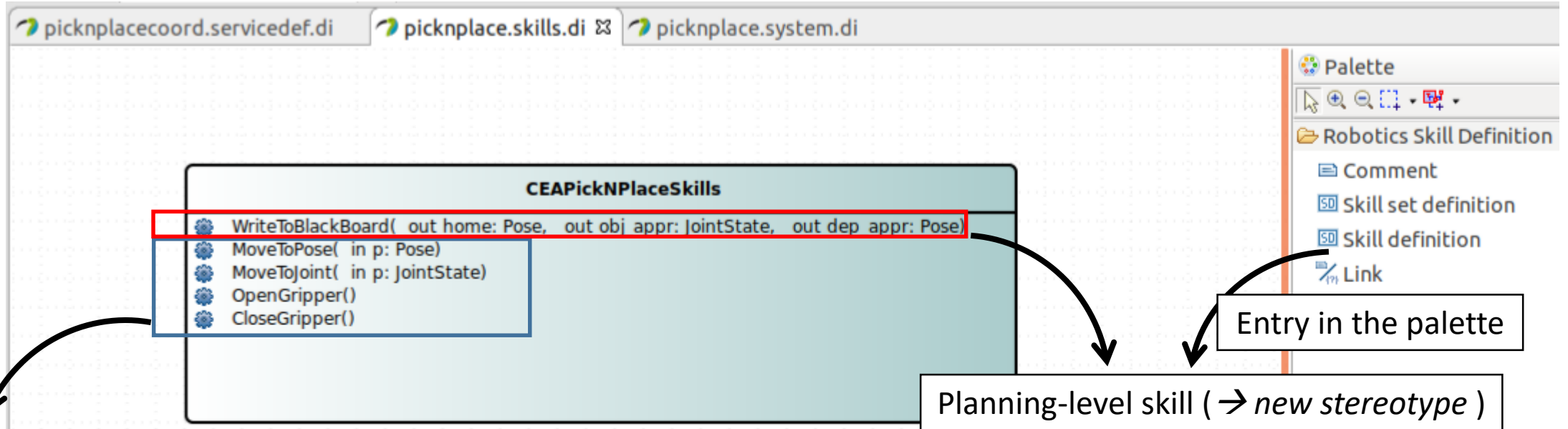


- Tasks, e.g., BTs (*→ uses skill signatures*)

Task  
Modeling



# Skill Definition — “Signatures”



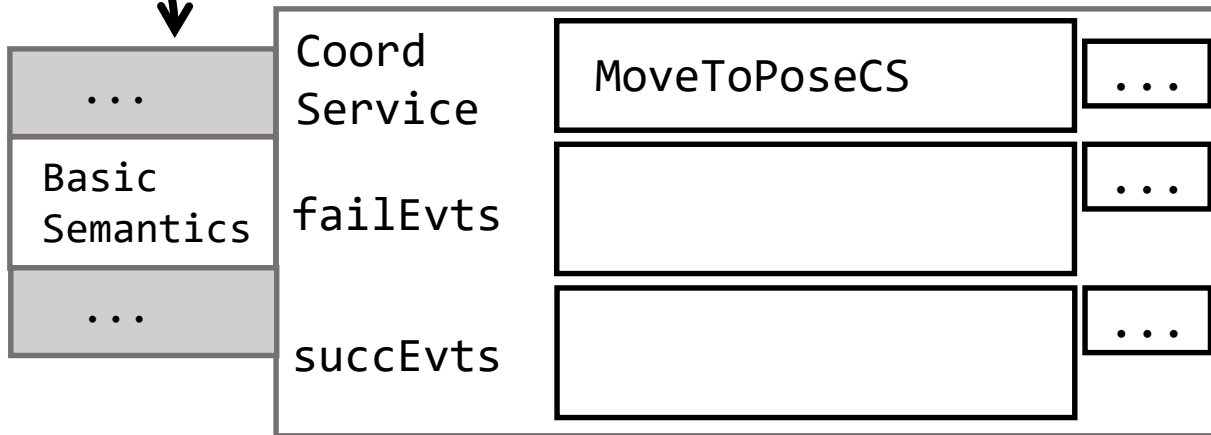
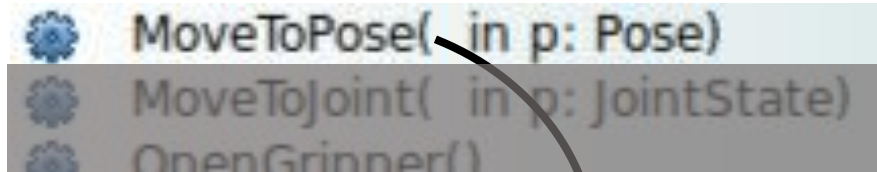
Skill realized by components.  
Skill code is **completely generated**  
as part of sequencer’s master  
coordination interface

Planning-level skill ( → *new stereotype* )  
Code structure is generated,  
code logics is provided by users  
**Temporary workaround until the link  
with planning level is not implemented**

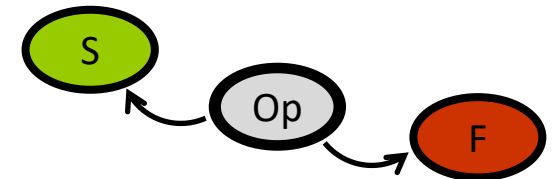
Entry in the palette

# Skill Definition — *Realization semantics (e.g., with FSMs)*

(1) Click → go to property view

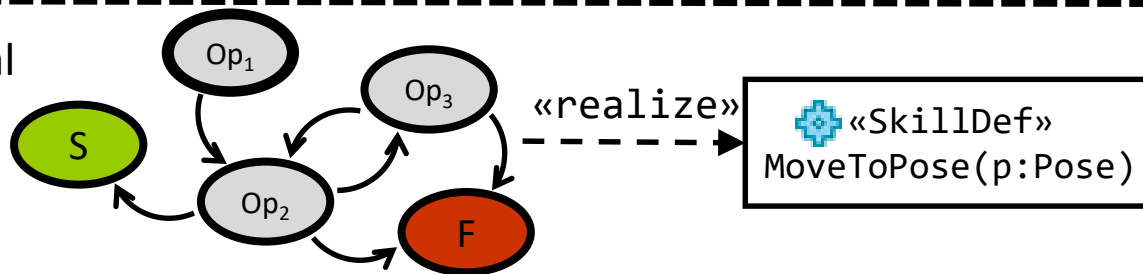


(2) is used to configure the simplest (basic) realization semantics for a skill (the user does not need to see it)



- ***MoveToPoseBasicFSM*** (UML model)
- ***CoordService*** describes the interaction of gray state with realizing component(s)
- ***failEvts*** (optional) describe the events causing transition to the red state
- ***succEvts*** (optional) describe the events causing transition to the green state

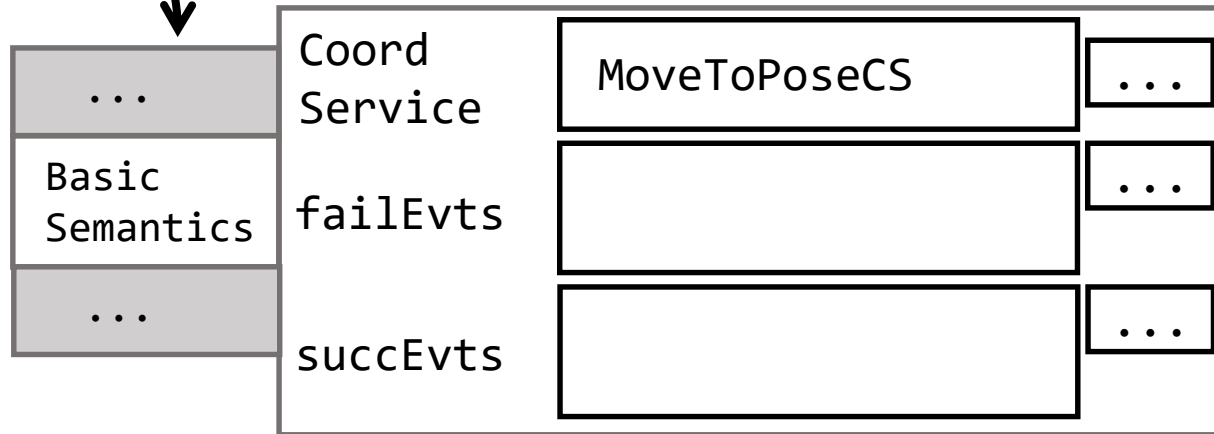
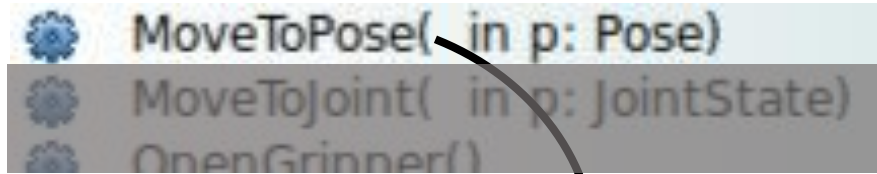
There may exist additional domain-specific realization for the skill.



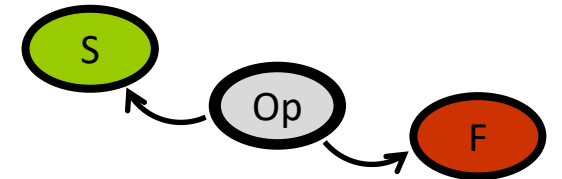
- Each realization is a FSM.
- Each FSM is designed in a dedicated view (still to be done)
- From this detailed view, *CoordServices* are indicated for each gray state, *evts* for transitions, etc.
- The selection of 1 specific realization for a skill definition is done later by the system builder

# (Not Yet Supported For the Next Release)

(1) Click → go to property view

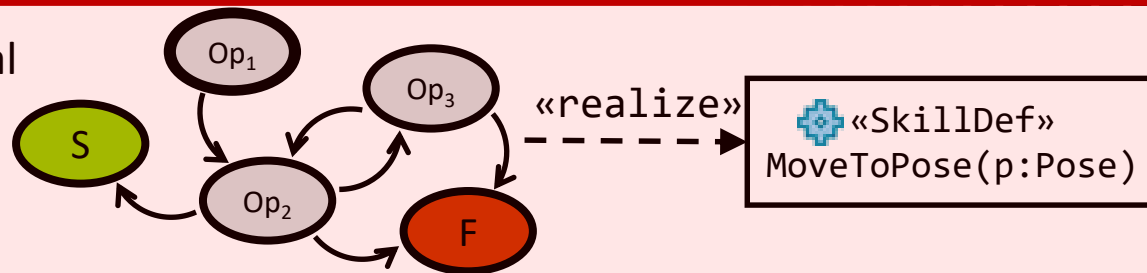


(2) is used to configure the simplest (basic) realization semantics for a skill (the user does not need to see it)



- **MoveToPoseBasicFSM** (UML model)
- **CoordService** describes the interaction of gray state with realizing component(s)
- **failEvs** (optional) describe the events causing transition to the red state
- **succEvs** (optional) describe the events causing transition to the green state

There may exist additional domain-specific realization for the skill.



- Each realization is a FSM.
- Each FSM is designed in a dedicated view (still to be done)
- From this detailed view, *CoordServices* are indicated for each gray state, *evts* for transitions, etc.
- The selection of 1 specific realization for a skill definition is done later by the system builder

# Roles, Models and their Usage

- **Coordination Service Definitions**

- Signatures—e.g., GoTo(Location loc)
- Realization semantics, e.g., in terms of FSMs (*uses Coordination Service Definitions*)



Services  
Definition

Tier 2

Skill  
Definition



Tier 3

*Component  
Development  
phase*

*System  
Integration  
phase*

*Behavior  
Development  
phase*

- Bindings between *Coordination Service Definitions* and component internals



Component  
Definition

- Select skill FSMs which best fit his application
- Ensure that components come with the needed coordination interfaces



System  
Build

- Tasks, e.g., BTs (*→ uses skill signatures*)

Task  
Modeling



code-generation, ...



# Binding of Coordination Services and Component Internals

(slave) coordination ports—  
provide coordination services

Coordination ports are bound  
to activities

Coordination ports → new  
stereotype ?

Each activity has functions to implement  
the coordination pattern provided by the  
port it is connected to (here *goal accept.*,  
*goal exec.*, *cancel action*; other functions  
could be *feedback*, *result comm.* etc.)

No changes!

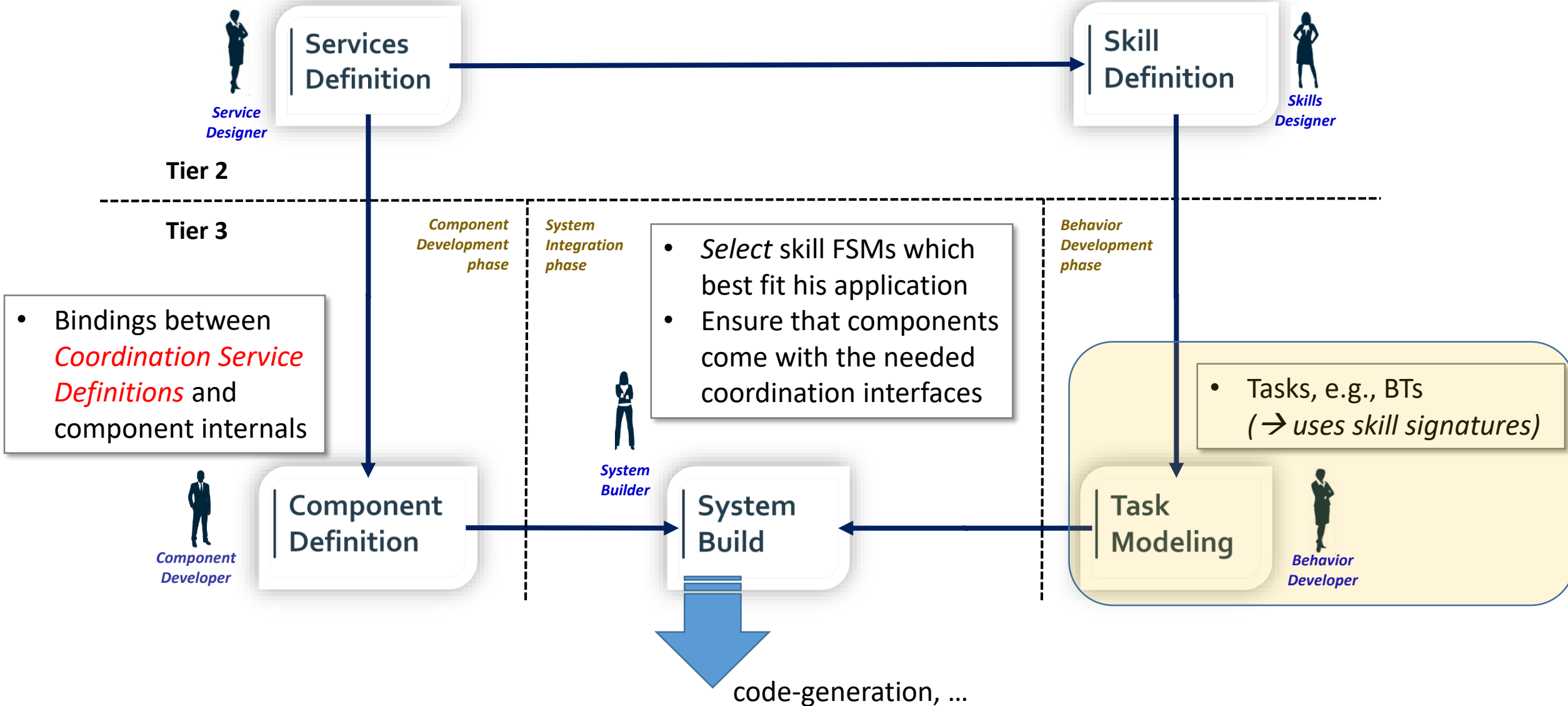
Model Explorer

- ComponentDefinitionModel IsybotCtrlCompDef
  - IsybotCtrl diagram
    - Package Import UML Primitive Types
    - Package Import robotics
    - Package Import PickNPlaceCoordServicedef
    - ComponentDefinition IsybotCtrl
      - Parameter Class
      - Activity Activity1
        - Activity Activity2
        - Activity Activity3
        - Activity Activity4
          - ActivityInstance activity1 : Activity1
          - ActivityInstance activity1 : Activity2
          - ActivityInstance activity2 : Activity3
          - ActivityInstance activity3 : Activity4
          - moveToCartePos provides MoveToPoseCS
          - moveToJointPos provides MoveToJointStateCS
          - openGripper provides OpenGripperCS
          - closeGripper provides CloseGripperCS
          - Pose, Empty, Empty
          - JointState, Empty, Empty
          - Empty, Empty, Empty
          - Empty, Empty, Empty
          - Function moveToCartePos\_goal
          - Function moveToCartePos\_cancel
          - Function moveToCartePos\_accepted
          - Function moveToJointPos\_goal
          - Function moveToJointPos\_cancel
          - Function moveToJointPos\_accepted
          - Function openGripper\_goal
          - Function openGripper\_cancel
          - Function openGripper\_accepted
          - Function closeGripper\_goal
          - Function closeGripper\_cancel
          - Function closeGripper\_accepted

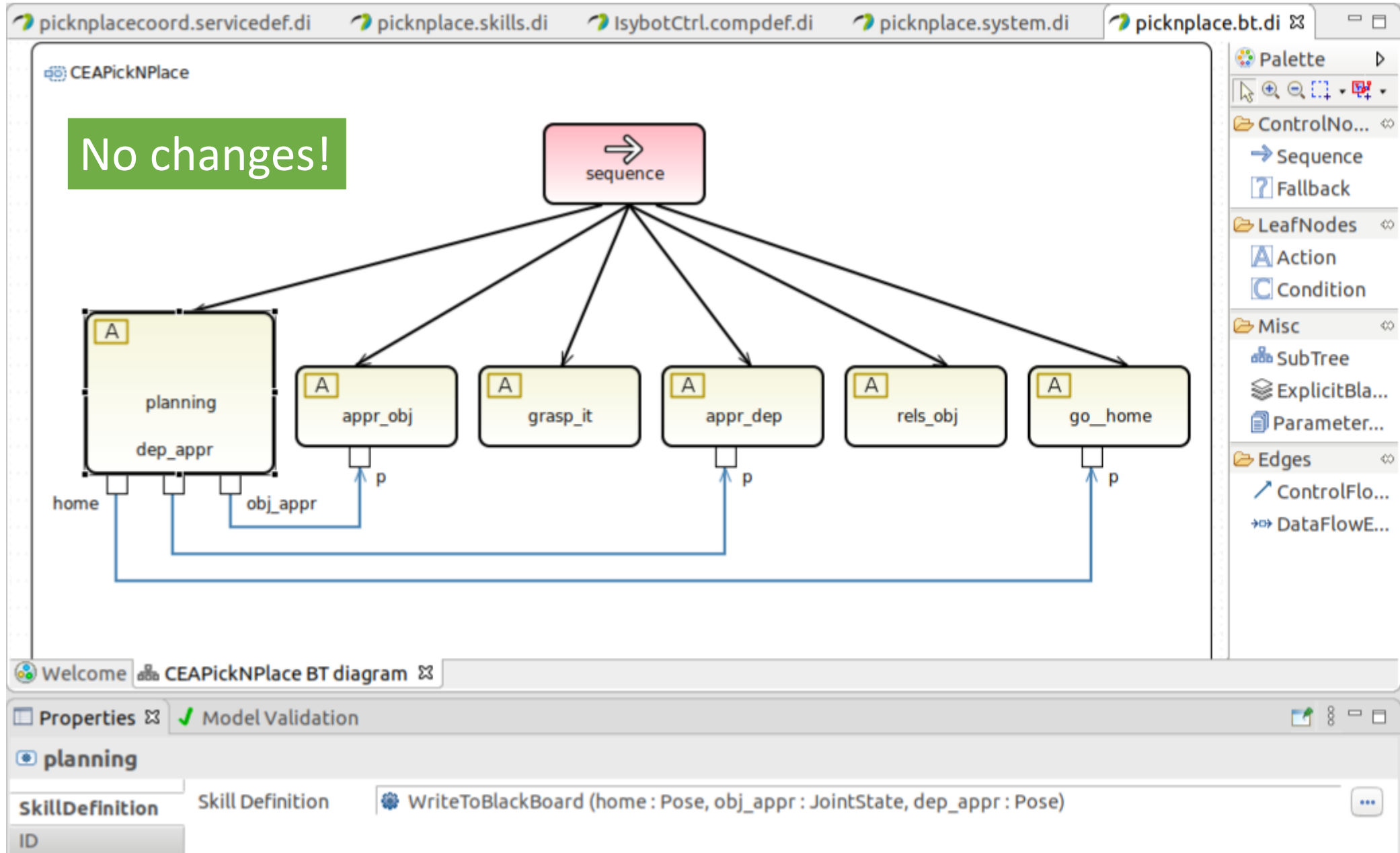
# Roles, Models and their Usage

- **Coordination Service Definitions**

- Signatures—e.g., GoTo(Location loc)
- Realization semantics, e.g., in terms of FSMs  
(*uses Coordination Service Definitions*)



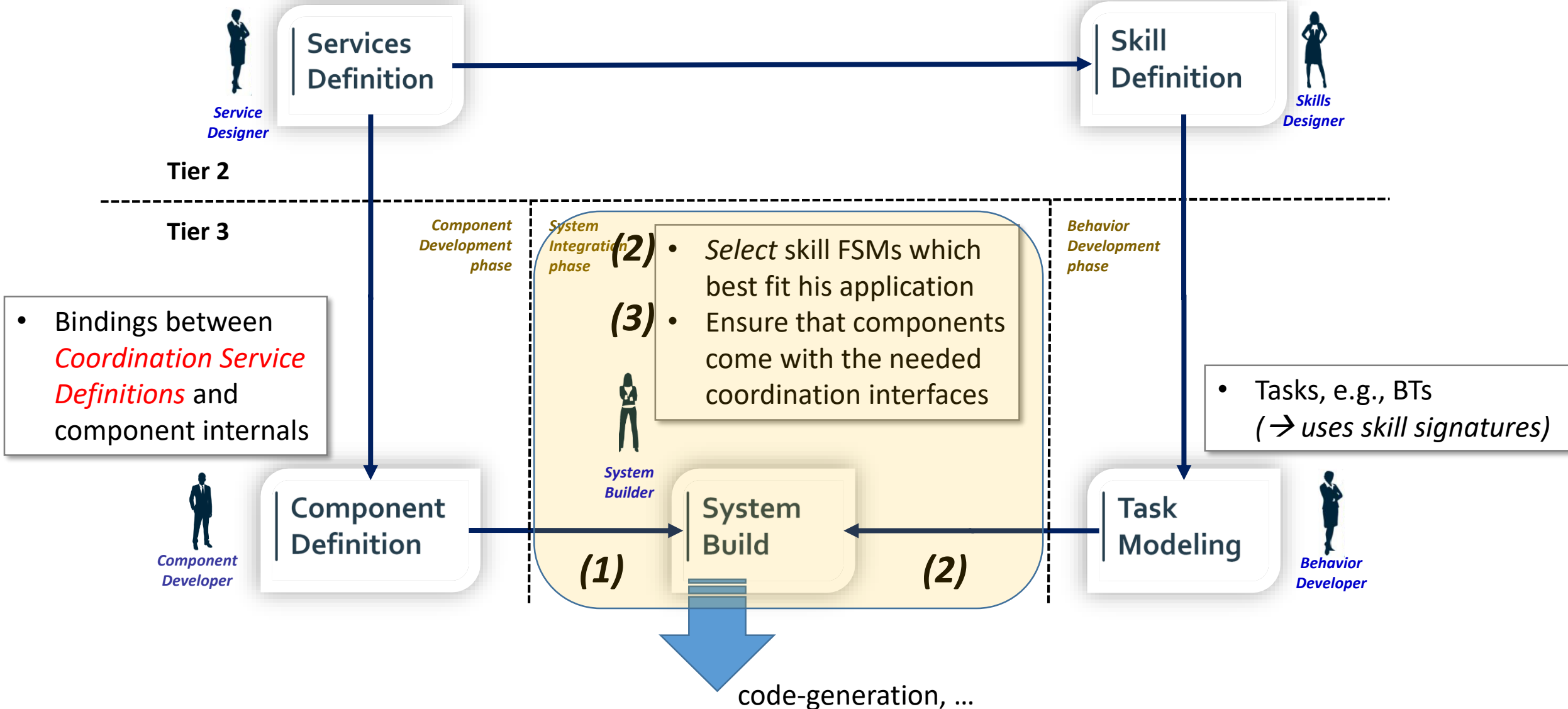
# Tasks (BTs)



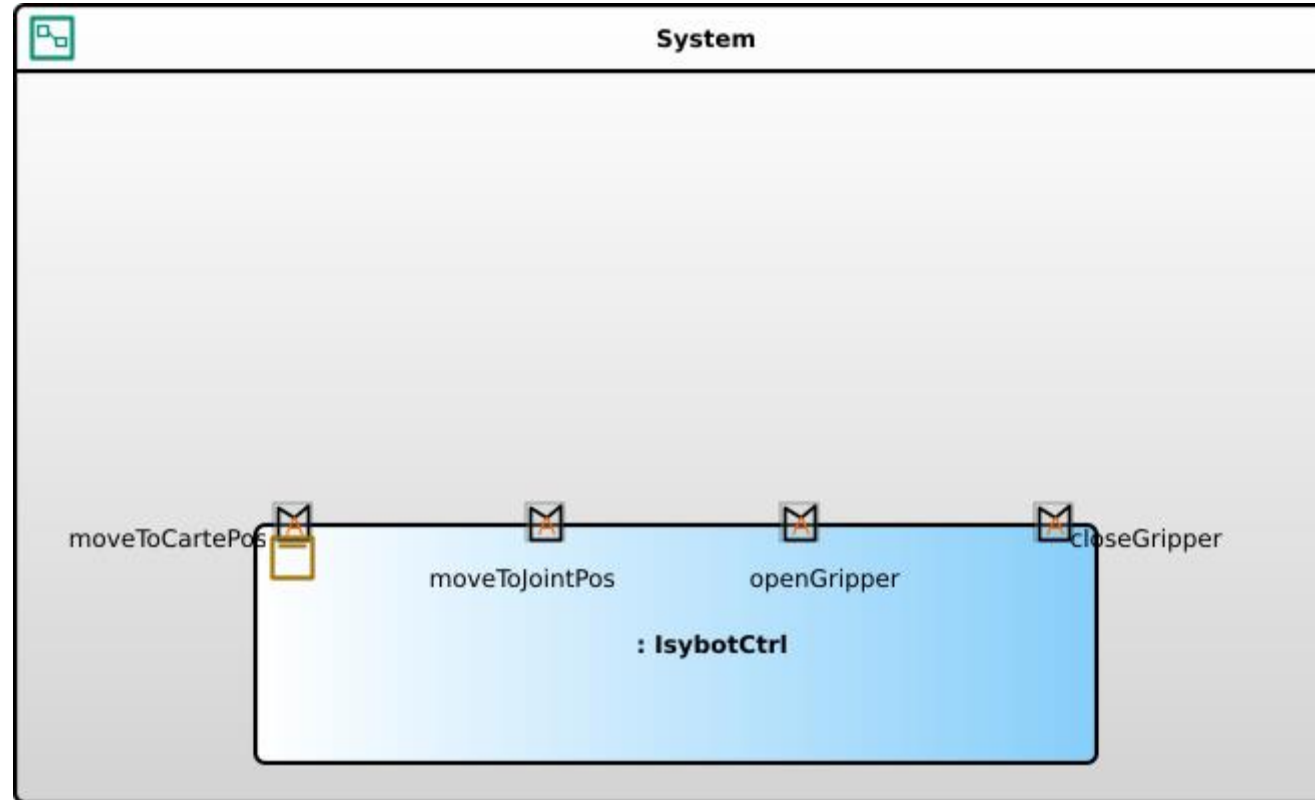
# Roles, Models and their Usage

- **Coordination Service Definitions**

- Signatures—e.g., GoTo(Location loc)
- Realization semantics, e.g., in terms of FSMs (*uses Coordination Service Definitions*)

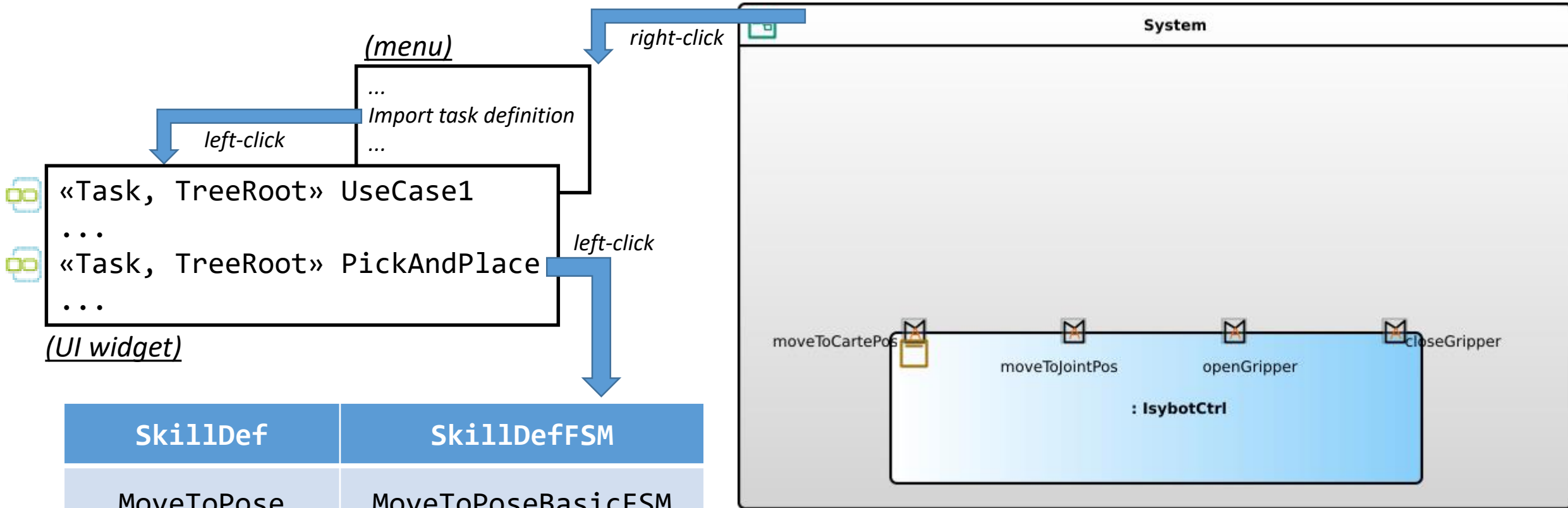


# System Build (1) — Components' Assembly



No changes!

# System Build (2) —Task Import and FSM Selection

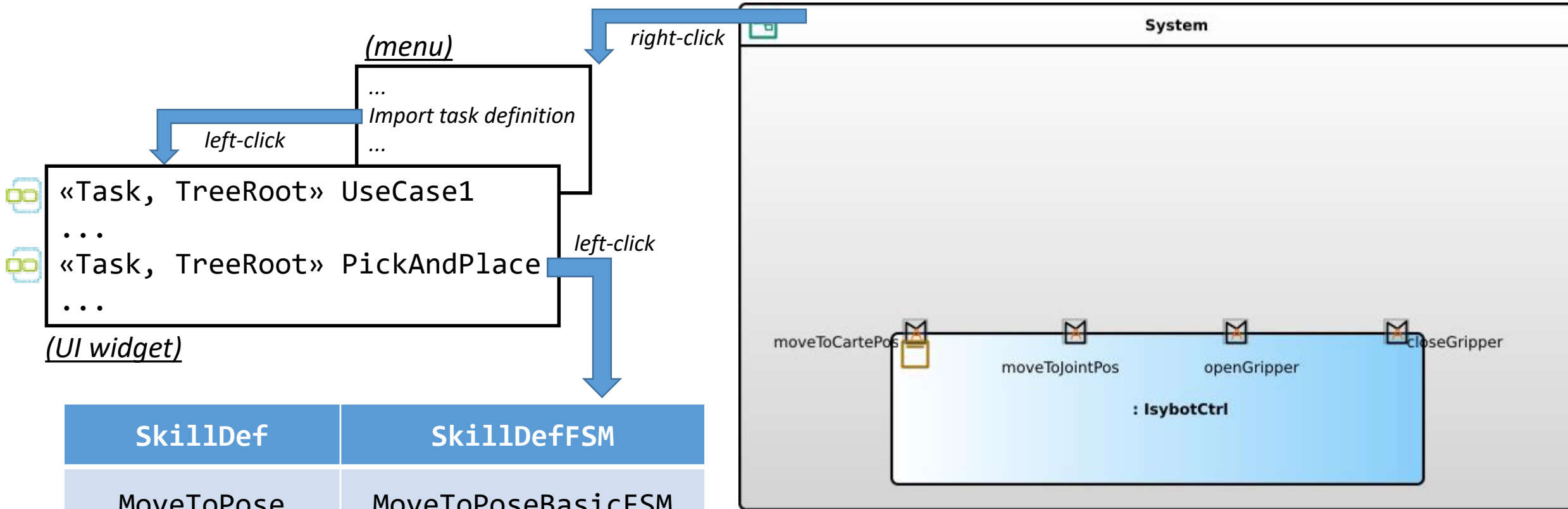


| SkillDef     | SkillDefFSM          |
|--------------|----------------------|
| MoveToPose   | MoveToPoseBasicFSM   |
| MoveToJoint  | MoveToJointBasicFSM  |
| OpenGripper  | OpenGripperBasicFSM  |
| CloseGripper | CloseGripperBasicFSM |

(Table)

If multiple “realize” relations are found for a skill definition, then the cell is empty. It is up to the user to select the FSM which best fits the operation context.

# (Not Yet Supported For the Next Release)



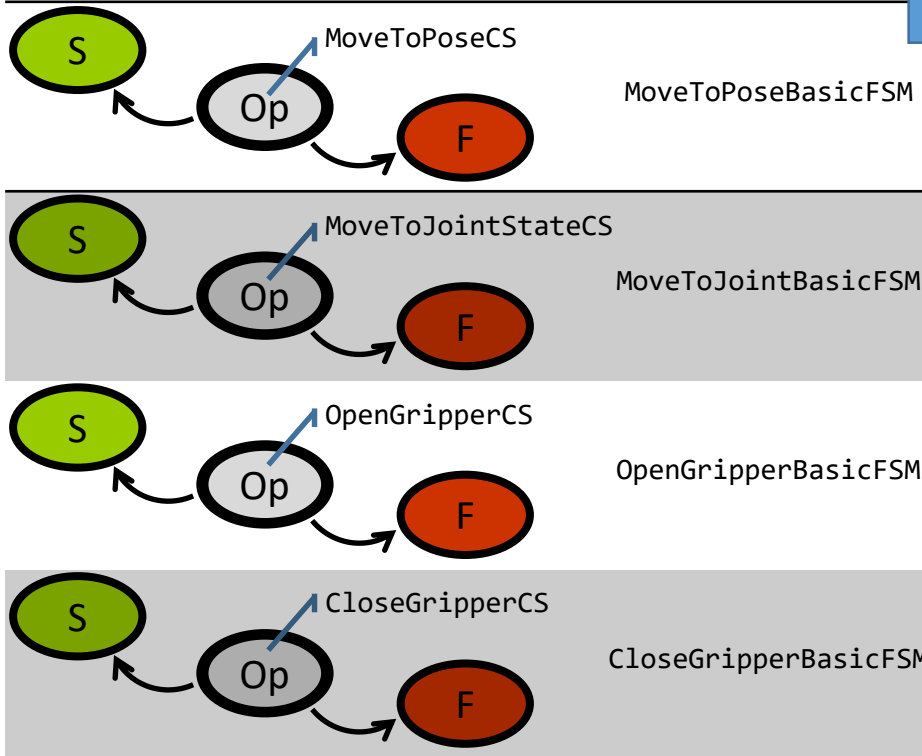
| SkillDef     | SkillDefFSM          |
|--------------|----------------------|
| MoveToPose   | MoveToPoseBasicFSM   |
| MoveToJoint  | MoveToJointBasicFSM  |
| OpenGripper  | OpenGripperBasicFSM  |
| CloseGripper | CloseGripperBasicFSM |

(Table)

If multiple "realize" relations are found for a skill definition, then the cell is empty. It is up to the user to select the FSM which best fits the operation context.

# System Build (3) — Sequencer and Interface Conformity

| SkillDef     | SkillDefFSM          |
|--------------|----------------------|
| MoveToPose   | MoveToPoseBasicFSM   |
| MoveToJoint  | MoveToJointBasicFSM  |
| OpenGripper  | OpenGripperBasicFSM  |
| CloseGripper | CloseGripperBasicFSM |



1 part for Op State  
(SkillDef name + 'Action')

